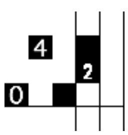
Steven R. Dunbar
Department of Mathematics
203 Avery Hall
University of Nebraska-Lincoln
Lincoln, NE 68588-0130
http://www.math.unl.edu
Voice: 402-472-3731
Fax: 402-472-8466

Topics in Probability Theory and Stochastic Processes Steven R. Dunbar

---

Distinguishing A Biased Coin From a Fair Coin



Rating

Mathematically Mature: may contain mathematics beyond calculus with proofs.
Mathematicians Only: prolonged scenes of intense rigor.

Section Starter Question

You have two identical coins except one is fair and the other has a biased chance $p > \frac{1}{2}$ of coming up heads. Unfortunately, you don't know which is which. How can you find the biased coin?

Key Concepts

1. How to numerically compute the probability of a majority of heads in a sequence of paired coin flips of a biased coin and a fair coin.

2. How to numerically compute the probability of statistical evidence of a biased coin in a sequence of paired coin flips of a biased coin and a fair coin.

3. Bayesian analysis using likelihood ratios as statistical evidence of a biased coin in a sequence of paired coin flips of a biased coin and a fair coin.

4. A theoretical analysis using probabilistic inequalities shows a necessary and sufficient number of flips is of the order $(p - 1/2)^{-2}$.

Vocabulary

1. A sequence of independent Bernoulli trials with probability 1/2 of success on each trial is metaphorically called a **fair** coin. One for which the probability is not 1/2 is a **biased** or **unfair** coin.

2. Suppose a fair coin 1 is in the left hand and a biased coin 2 is in the right hand. The coin in the left hand comes up heads k_L times and the coin in the right hand comes up heads k_R times. Let $k = (k_L, k_R)$ and call this scenario θ_L. The **likelihood** of this scenario occurring is

$$Q(k \mid \theta_L) = B(k_L, n, p_1) B(k_R, n, p_2).$$

3. If fair coin 1 is in the right hand and biased coin 2 is in the left hand then call this scenario θ_R. The **likelihood ratio** is

$$\frac{Q(k \mid \theta_L)}{Q(k \mid \theta_R)}.$$

Mathematical Ideas

A Simple Probability Problem

This problem serves as a warm-up for the more detailed investigations below.

You have two coins. One is fair with $\mathbb{P}[H] = \frac{1}{2}$. The other coin is biased with $\mathbb{P}[H] = \frac{2}{3}$. First you toss one of the coins once, resulting in heads. Then you toss the other coin three times, resulting in two heads. Which coin is more likely to be the biased coin, the first or the second?

Joint Probability Solution

Assuming tossing the biased coin once and tossing the fair coin three times, the probability of observing the outcome is

$$\frac{2}{3} \cdot \binom{3}{2} \left(\frac{1}{2}\right)^2 \left(\frac{1}{2}\right)^1 = \frac{2}{8}.$$

On the other hand, assuming tossing the fair coin once and tossing the biased coin three times, the probability of observing the given outcome is

$$\frac{1}{2} \cdot \binom{3}{2} \left(\frac{2}{3}\right)^2 \left(\frac{1}{3}\right)^1 = \frac{2}{9}.$$

This means it was more likely the biased coin was tossed once while the fair coin was tossed three times.

Using Bayes Theorem

Take a uniform prior, that is, assume the fair coin is equally likely to be either of the two coins. Let $K = (k_1, k_2)$ denote the observation, and B the event that the first coin is biased. Then

$$\mathbb{P}\left[B\right] = \mathbb{P}\left[B^C\right] = \frac{1}{2}.$$

The conditional probabilities are

$$\mathbb{P}\left[K \mid B\right] = \frac{2}{3} \cdot \binom{3}{2} \left(\frac{1}{2}\right)^2 \left(\frac{1}{2}\right)^1 = \frac{2}{8}$$

and

$$\mathbb{P}\left[K \mid B^C\right] = \frac{1}{2} \cdot \binom{3}{2} \left(\frac{2}{3}\right)^2 \left(\frac{1}{3}\right)^1 = \frac{2}{9}.$$

From Bayes Theorem,

$$\begin{aligned}
\mathbb{P}\left[B \mid K\right] &= \frac{\mathbb{P}\left[K \mid B\right] \cdot \mathbb{P}\left[B\right]}{\mathbb{P}\left[K \mid B\right] \cdot \mathbb{P}\left[B\right] + \mathbb{P}\left[K \mid B^C\right] \cdot \mathbb{P}\left[B^C\right]} \\
&= \frac{2/8}{2/8 + 2/9} = \frac{9}{17} > \frac{1}{2}.
\end{aligned}$$

Using Likelihood Ratios

Take even prior odds to be even, that is, assume the fair coin is equally likely to be either of the two coins. Let $K = (k_1, k_2)$ be the observed outcomes. The likelihood ratio is likelihood ratio]

$$\begin{aligned}
L(K) &= \frac{\mathbb{P}\left[K \mid B\right]}{\mathbb{P}\left[K \mid B^C\right]} \\
&= \frac{\frac{2}{3}\binom{3}{2}\left(\frac{1}{2}\right)^3}{\frac{1}{2} \cdot \binom{3}{2}\left(\frac{2}{3}\right)^2\left(\frac{1}{3}\right)^1} \\
&= \frac{6/24}{12/54} = \frac{9}{8}.
\end{aligned}$$

Then the odds in favor of the first coin being biased and the second coin being fair are $9:8$, so the probability is $\frac{9}{8+9} = \frac{9}{17}$.

A Larger Problem

The following problem appeared in the FiveThirtyEight.com weekly Riddler puzzle column on September 29, 2017:

> On the table in front of you are two coins. They look and feel identical, but you know one of them has been doctored. The fair coin comes up heads half the time while the doctored coin comes up heads 60 percent of the time. How many flips – you must flip both coins at once, one with each hand – would you need to give yourself a 95 percent chance of correctly identifying the doctored coin?
>
> Extra credit: What if, instead of 60 percent, the doctored coin came up heads some percent p of the time? How does that affect the speed with which you can correctly detect it?

This problem appeared in a paper "What's Past is *Not* Prologue" by James White, Jeff Rosenbluth, and Victor Haghani [**?**]. In turn, a problem posed in a paper "Good and bad properties of the Kelly criterion" by MacLean, Thorp and Ziemba [**?**] inspired the problem by White, Rosenbluth, and Haghani.

Solving this problem requires some interpretation and computation.

Probability of a Majority of Heads

A sequence of independent Bernoulli trials with probability $1/2$ of success on each trial is metaphorically called a **fair** coin. One for which the probability is not $1/2$ is a **biased** or **unfair** coin. Continuing the coin metaphor, The trials are often called flips.

Any fixed number of simultaneous flips always has a chance that the fair coin will have more heads than the biased coin. But the Weak Law of Large Numbers says the probability that the biased coin will have a majority of heads increases to 1 as the number of flips increases. One way to decide which is the biased coin is to choose the coin which has a majority of heads. The authors White, Rosenbluth, and Haghani want to calculate how many paired flips of 2 coins, one biased and one fair, we must observe in order to be 95% confident that the coin with more heads is the biased coin.

Assuming independence, since each coin has a binomial distribution the joint probability mass distribution after n flips is the product of the individual binomial probability mass distributions. Denote by $Q(n; j, k)$ the probability of j heads for the fair coin and k heads for the biased in n flips. Use p for the probability of heads of the biased coin and $\frac{1}{2}$ for the probability of heads for the fair coin. Then

$$Q(n; j, k) = \binom{n}{j} \left(\frac{1}{2}\right)^j \left(\frac{1}{2}\right)^{n-j} \cdot \binom{n}{k} p^k (1-p)^{n-k}.$$

Then summing over values where $j < k$ gives the probability that the biased coin has more heads than the fair coin in n flips

$$\mathbb{P}[j < k] = \frac{1}{2^n} \sum_{k \le n} \sum_{j<k} \binom{n}{j} \binom{n}{k} p^k (1-p)^{n-k}.$$

This sum has no closed formula evaluation so calculation is necessary. First create two vectors of length $n+1$ with the binomial distribution on 0 to n with probability $\frac{1}{2}$ for the fair coin and with probability p for the biased coin. Then using the outer product of these two vectors create the bivariate binomial distribution for the two coins. This will be an $(n+1) \times (n+1)$ matrix. The element in row i and column j is the product of the binomial probability of $i-1$ heads for the fair coin and the binomial probability of $j-1$ heads for the biased coin. The sum is over column indices strictly greater than the row indices. To create the sum, set the lower triangular part of the matrix and the diagonal of the matrix to 0 and use the sum command to sum all entries of the matrix. This seems to be efficient, even for values of n up to 200. To find the minimal value of n for which this probability is greater than 0.95 use binary search in n over a reasonable interval.

It might seem possible to use a bivariate normal approximation of the bivariate binomial distribution to calculate the probability. Although approximation of the bivariate binomial distribution with a bivariate normal distribution is possible, the double integration of the bivariate normal with a non-zero covariance would be over a region of the form $y > x - \epsilon$. The R language has no direct way to calculate the integral over a region of this form, so it is actually easier to calculate with the bivariate binomial distribution.

The result is that it takes 143 flips of the two coins for the probability to be greater than 0.95 for the 0.6 biased coin to have more heads than the fair coin.

The same analysis for various probabilities of heads p for the biased coin and for values 0.95, 0.9 and 0.75 of certainty is in Figure 1. The number of flips required decreases as the bias p increases, as expected. The number of flips required also decreases as the certainty decreases.

Using the Central Limit Theorem

Assume the biased coin is in the left hand. Let L_j be the result of left-hand coin flip j, knowing it is biased so $L_j = \text{Head} = 1$ with probability 0.6, $L_j = \text{Tail} = 0$ with probability 0.4. Let R_j be the result of right-hand coin flip j, knowing it is fair so $R_j = \text{Head} = 1$ with probability 0.5, $R_j = \text{Tail} = 0$ with probability 0.5. Let $X_j = L_j - R_j$. This is a trinomial random variable with $X_j = -1$ with probability $1/5 = 0.2$, $X_j = 0$ with probability $1/2 = 0.5$, $X_j = 1$ with probability $3/10 = 0.3$.

Consider the statistics of X_j,

$$\begin{aligned}
\mathbb{E}[X_j] &= (-1)\cdot(0.2) + 0\cdot(0.5) + (+1)\cdot(0.3) = 0.1, \\
\operatorname{Var}[X_j] &= (-1)^2\cdot(0.2) + (0)^2\cdot(0.5) + (+1)^2\cdot(0.3) - (0.1)^2 \\
&= 0.49, \\
\sigma[X_j] &= 0.7.
\end{aligned}$$

Let $S_n = \frac{1}{n}\sum_{j=1}^{n} X_j$ be the sample mean. The distribution of S_n is on -1 to 1 by increments of $1/n$ for a total of $2n+1$ points with

$$\begin{aligned}
\mathbb{E}[S_n] &= 0.1, \\
\operatorname{Var}[S_n] &= 0.49/n, \text{ by independence}, \\
\sigma[S_n] &= 0.7/\sqrt{n}.
\end{aligned}$$

Thus the distribution of S_n clusters around $1/10$ with standard deviation $7/(10\sqrt{n})$. If the biased coin is in the right hand, the distribution of S_n clusters around $-1/10$ with standard deviation $7/(10\sqrt{n})$.The goal is find a number of flips such that with high probability the sample mean S_n is closer to the theoretical mean 0.1 (when the biased coin is in the left hand) than the value -0.1 (the theoretical mean coming from the situation with the biased coin in the right hand). Take "the sample mean closer to 0.1 than -0.1" to mean S_n is closer to 0.1 than the distance to the midpoint 0 between the two means. Precisely, the goal is to find n so that $\mathbb{P}[S_n - 0.1 > -0.1] \geq 0.95$.

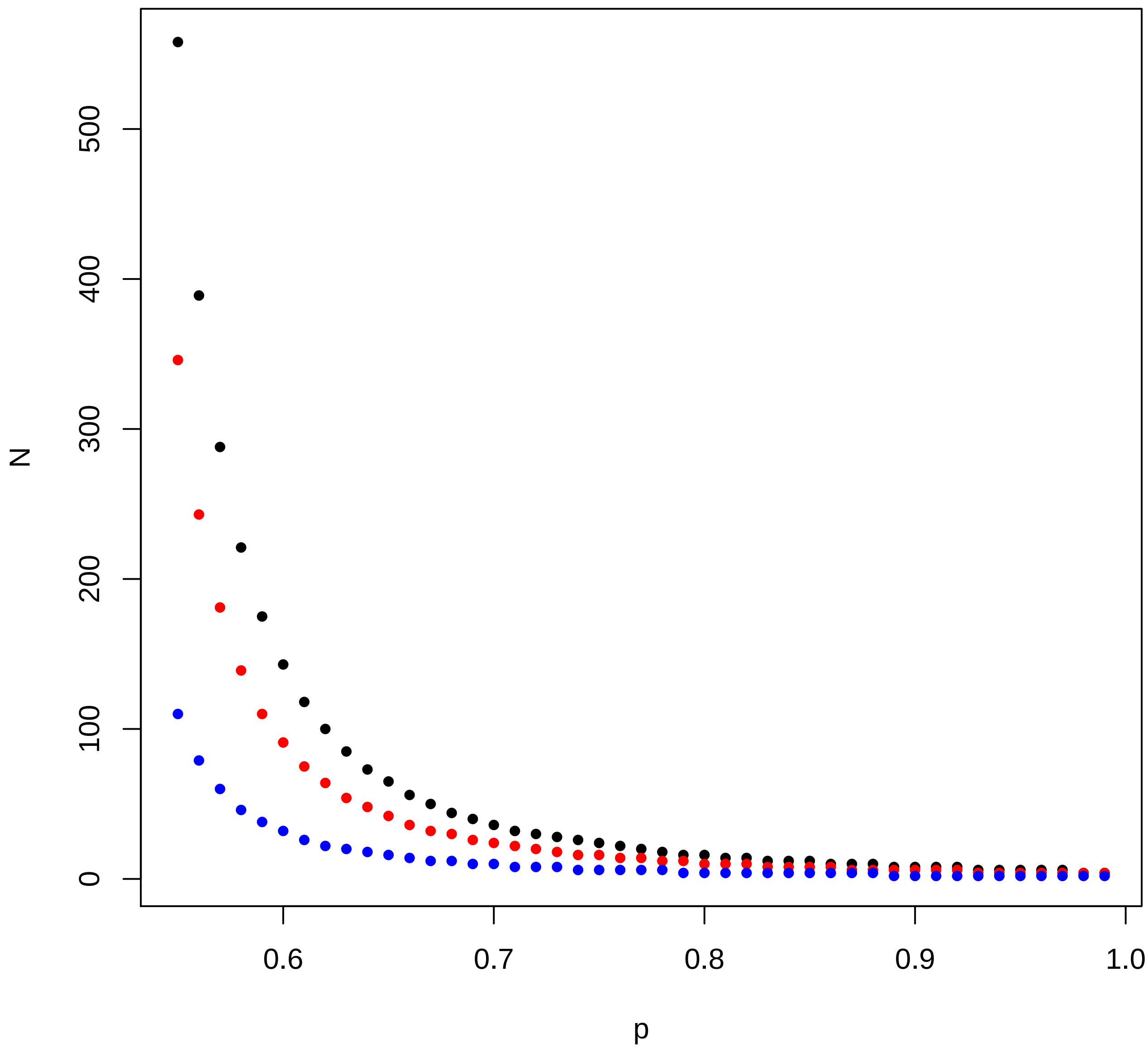


Figure 1: The number of flips N required to get a majority of heads with certainty 0.95 (black), 0.9 (red) and 0.75 (blue).

This is the same event as $\mathbb{P}[S_n > 0] = \mathbb{P}[L_n - R_n > 0] = \mathbb{P}[L_n > R_n]$ which is the same criterion as having the majority of heads above.

Expressing the precise distribution of S_n analytically is difficult. So instead, use a numerical calculation of the distribution. To numerically calculate the distribution of the sample mean S_n, use the R package `distr` and specifically the function `convpow` that takes the n-fold convolution power of a distribution to create the distribution of the n-fold sum of a random variable. Note that mathematically the support of the distribution of, for instance S_{100}, would be from -1 to 1, with 201 points. However, the actual calculated distribution support of, for instance, S_{100}, is 111 points from -46 to 64. The reason is that `convpow` ignores points with probability less than 10^{-16} and so these points are not included in the domain. So use `match(0, (support(D100)))` to find the index of 0. This turns out to be index 47. So summing the distribution over indices from $47+1$ to 111 gives the probability of the random variable $S_n > 0$. Searching for a value of n large enough that the probability exceeds 0.95 gives the required number of flips. From some experimentation, the numerical support of the distribution is positive for $n \geq 150$, that is, $\mathbb{P}[S_n \geq 0] = 1$ for $n \geq 150$. That means that the numerical search should start from a high of 150.

Using this algorithm, the necessary number of flips to distinguish a biased coin with a probability of heads 0.6 from a fair coin with a certainty level of 0.95 is 143. The same analysis for various probabilities of heads p for the biased coin and for values 0.95, 0.9 and 0.75 of certainty is in Figure 2. The number of flips required decreases as the bias increases as expected. The number of flips required also decreases to 5 for certainty 0.95, 4 for certainty 0.9 and 3 for certainty 0.75.

Connections between the Two Calculations

The calculation of the probabilities uses fundamentally the same information, as the diagram in Figure 3 for the case $n = 10$ illustrates. The 11×11 array of dots represents the support of the *bivariate binomial distribution* as a matrix, with rows from 0 to 11 for the number of heads from the fair coin and columns 0 to 11 for number of heads for the biased coin.

The probability that the majority of flips is from the biased coin is the sum of the probabilities in the strict upper triangle of the support of the bivariate distribution. The shaded part of Figure 3 shows this domain.

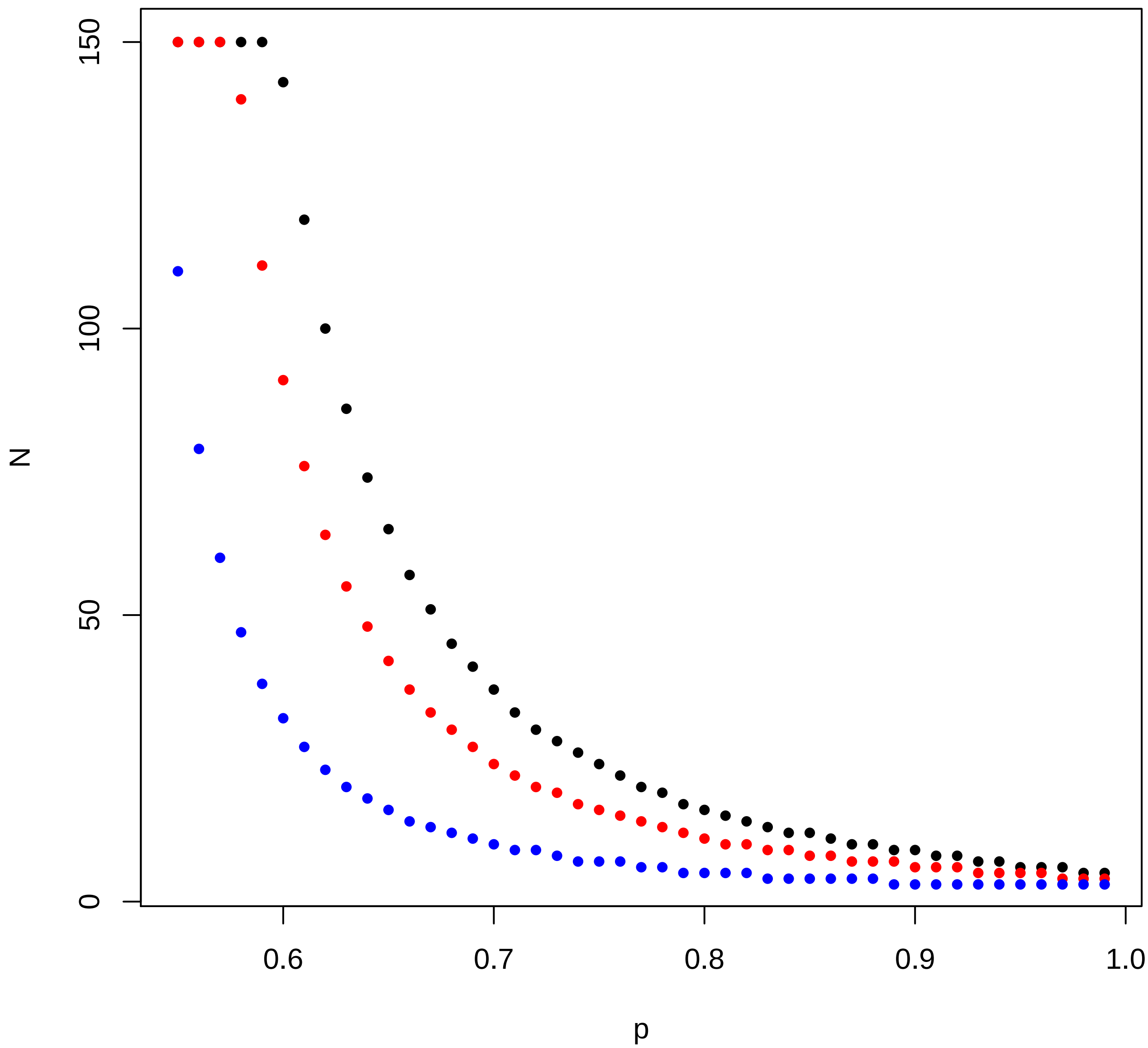


Figure 2: The number of flips N required to statistically distinguish a coin with probability p from a fair coin with 0.95 (black), 0.9 (red) and 0.75 (blue).

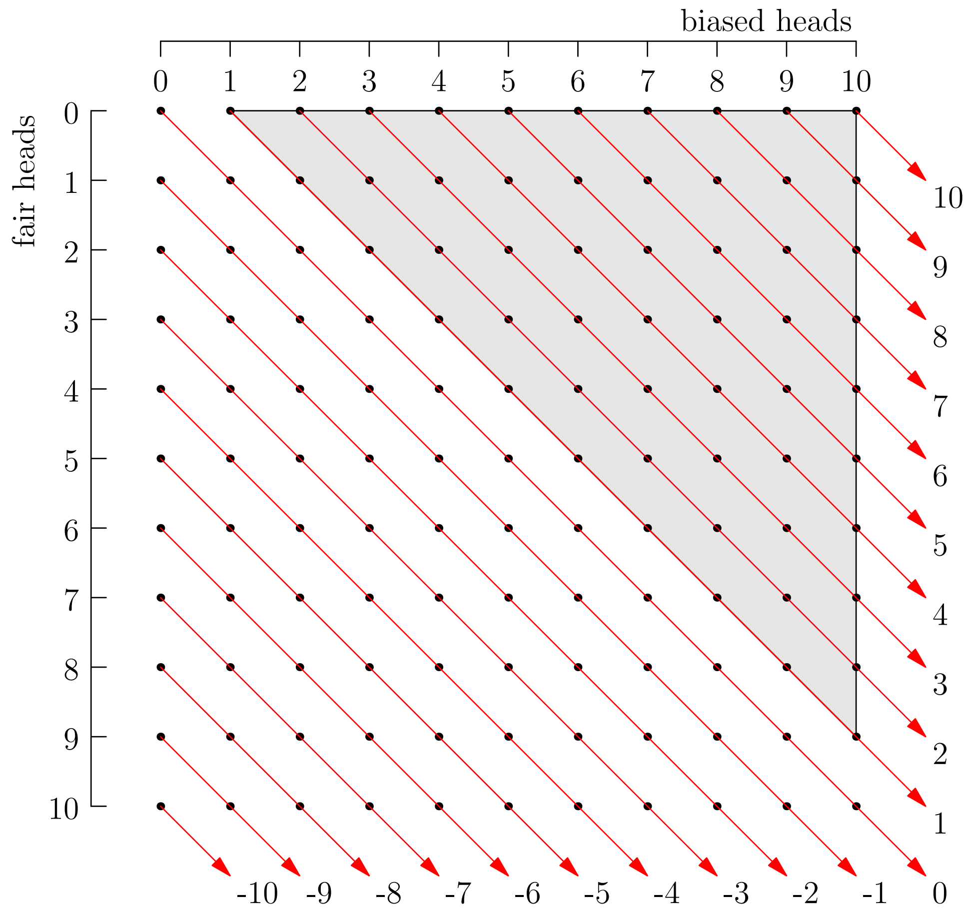


Figure 3: The support of the bivariate binomial and the calculation of the multinomial distribution for $n = 10$.

The *multinomial probability distribution* on -10 to 10 of the sum $\sum_{j=0}^{10}(L_j - R_j)$ is the sum of the probabilities along diagonals of the bivariate binomial distribution as indicated by the red arrows. In the case the biased coin has $p = 0.6$ so the mean of $L_n - R_n = 0.1$, the probability of the event

$$\begin{aligned} [S_n > 0] &= \left[\frac{1}{n}\sum_{j=1}^{n}(L_j - R_j) > 0\right] \\ &= \left[\sum_{j=1}^{n}(L_j - R_j) > 0\right] \\ &= \left[\sum_{j=1}^{n} L_j > \sum_{j=1}^{n} R_j\right] \end{aligned}$$

is the sum of the probabilities in the strict upper triangle. This is the same criterion as for the majority of heads above. For $p = 0.6$ the search using the sample mean again requires 143 pair flips to distinguish the coins with certainty 0.95.

The framing of the application motivates the choice of criterion. The original problem posed in FiveThirtyEight.com comes from a white paper by James White, Jeff Rosenbluth, and Victor Haghani from Elm Partners Investing. The paper "Good and bad properties of the Kelly criterion" by McLean, Thorp and Ziemba motivated their example. The question is a simplified and idealized version of an investment question about how to identify a higher performing investment, modeled by a biased coin, from a lower performing investment with just an even chance at making a profit, modeled by a fair coin. In that case, the 95% certainty of a majority of gains is a reasonable choice of criterion. If the question is merely to identify the biased coin, then it makes sense to use the Central Limit Theorem to distinguish the mean given that the biased coin is in the left hand from the mean given that the biased coin is in the right hand.

Bayesian Solution

The coin flips are independent. So if a coin comes up heads with probability p and we flip it n times, the binomial distribution $B(k, n, p) = \binom{n}{k} p^k (1-p)^{n-k}$ gives the probability that it comes up heads exactly k times. Number the coins 1 and 2 probabilities of coming up heads p_1 and p_2 respectively. Flip each coin n times and record how many times each coin come up heads, but we don't know which coin is which. Say the coins come up heads k_L and k_R times for the left and right coin respectively. Call the vector of observed data $k = (k_L, k_R)$.

Exactly two possible scenarios occur:

Fair on Left Fair coin 1 was in the left hand and biased coin 2 was in the right hand and the observed outcome is $k = (k_L, k_R)$. Call this scenario θ_L. The **likelihood** of this scenario occurring is

$$Q(k \mid \theta_L) = B(k_L, n, p_1)B(k_R, n, p_2).$$

Fair on Right Fair coin 1 was in the right hand and biased coin 2 was in the left hand and the observed outcome is $k = (k_L, k_R)$. Call this scenario θ_R. The likelihood of this scenario occurring is

$$Q(k \mid \theta_R) = B(k_L, n, p_2)B(k_R, n, p_1).$$

If fair coin 1 is in the right hand and biased coin 2 is in the left hand then call this scenario θ_R. The **likelihood ratio** is

$$\frac{Q(k \mid \theta_L)}{Q(k \mid \theta_R)}.$$

Compute the posterior probability by using Bayes' rule:

$$Q(\theta_L \mid k) = \frac{Q(k \mid \theta_L) \cdot \mathbb{P}[\theta_L]}{Q(k \mid \theta_L) \cdot \mathbb{P}[\theta_L] + Q(k \mid \theta_R) \cdot \mathbb{P}[\theta_R]}$$

and similarly for $Q(\theta_R \mid k)$. Assume scenarios θ_L and θ_R have the same prior probability so $\mathbb{P}[\theta_L] = \mathbb{P}[\theta_R] = 1/2$. Therefore, the Bayesian condition for confidence simplifies to

$$\frac{Q(k \mid \theta_L)}{Q(k \mid \theta_L) + Q(k \mid \theta_R)} > 1 - \alpha \text{ or } \frac{Q(k \mid \theta_R)}{Q(k \mid \theta_L) + Q(k \mid \theta_R)} > 1 - \alpha.$$

To be confident with level α about one of the scenarios then the posterior probability must be greater than $1 - \alpha$.

Fair on Left If $Q(\theta_L \mid k) > 1 - \alpha$, then we are confident in Scenario Fair on Left.

Fair on Right If $Q(\theta_R \mid k) > 1 - \alpha$, then we are confident in Scenario Fair on Right.

By definition, $Q(\theta_L \mid k) + Q(\theta_R \mid k) = 1$ for all k. So the two cases above can't simultaneously be true when $\alpha < 1/2$. Rearranging these inequalities, we have the likelihood ratio

$$\frac{Q(k \mid \theta_L)}{Q(k \mid \theta_R)} > \frac{1-\alpha}{\alpha} \text{ or } \frac{Q(k \mid \theta_R)}{Q(k \mid \theta_L)} > \frac{1-\alpha}{\alpha}.$$

The implication is that we can stop flipping coins once the difference between log-likelihoods grows sufficiently large. The smaller is α, the larger the difference in log-likelihoods must be before we are confident. Simplify the expression by substituting the definition for the likelihoods $Q(k \mid \theta_L)$ and $Q(k \mid \theta_R)$ in terms of the binomial probabilities becoming

$$\frac{Q(k \mid \theta_L)}{Q(k \mid \theta_R)} = \left(\frac{p_1}{p_2}\right)^{k_L - k_R} \cdot \left(\frac{1-p_2}{1-p_1}\right)^{k_L - k_R}.$$

Take logarithms of both sides and combine the expressions into a single inequality involving the absolute difference of log-likelihoods

$$|\log Q(k \mid \theta_L) - \log Q(k \mid \theta_R)| > \log\left(\frac{1}{\alpha} - 1\right).$$

The implication is that we can stop flipping coins once the difference between log-likelihoods grows sufficiently large. The smaller is α, the larger the difference in log-likelihoods must be before we can declare that we are confident. After simplification, this becomes

$$|k_L - k_R| \cdot |\log\left(\frac{1}{p_1} - 1\right) - \log\left(\frac{1}{p_2} - 1\right)| > \log\left(\frac{1}{\alpha} - 1\right).$$

The number of heads k_L and k_R in the left and right hands are random variables. In one hand the number of heads from the coin with p_1 will have the binomial probability distribution with mean np_1 and standard deviation $\sqrt{np_1(1-p_1)}$. In the other hand the number of heads from the coin with p_1

will have the binomial probability distribution with mean np_2 and standard deviation $\sqrt{np_2(1-p_2)}$. Normal distributions with corresponding means and standard deviations can approximate each binomial distribution.

As a first approximation, substitute the empirical probabilities $\hat{p}_1 = k_L/n$ and $\hat{p}_2 = k_R/n$ obtained from the distribution means. Rearrange to isolate n to obtain

$$n > \frac{1}{|\hat{p}_1 - \hat{p}_2|} \cdot \frac{\log\left(\frac{1}{\alpha} - 1\right)}{|\log\left(\frac{1}{p_1} - 1\right) - \log\left(\frac{1}{p_2} - 1\right)|}.$$

This expression is a lower bound on the number of samples required, in terms of the confidence α, the known probabilities p_1 and p_2 and the empirical probabilities $\hat{p}_1$ and $\hat{p}_2$. Note that if $p_1 - p_2 \to 0$, then $n \to \infty$ which is expected. Taking $\hat{p}_1 = p_1 = 0.5$, $\hat{p}_2 = p_2 = 0.6$ and $\alpha = 0.05$, then $n = 72.619$. More generally, Figure 4 shows a plot of the number of flips required as a function of p_2 for fixed $p_1 = 0.5$ and fixed $\alpha = 0.05$, so in percentages, 95% confidence.

However, this is only a lower bound on the real number of flips required to find the biased coin since k_L and k_R can be closer than what was used above. In fact, in the scenario Fair on Left can have k_L in the range $[0, np_1 + z_\beta\sqrt{np_1(1-p_1)}]$ and k_R in the range $[np_2 - z_\beta\sqrt{np_2(1-p_2)}, n]$. The probability of each of these events is determined by quantiles z_β of the binomial distribution, or approximately by the normal distribution. Since $\sqrt{np_2(1-p_2)} < \sqrt{n \cdot \frac{1}{2} \cdot \frac{1}{2}} = \frac{1}{2}\sqrt{n}$

$$\begin{aligned}
|k_L - k_R| &> (np_2 - z_\beta\sqrt{np_2(1-p_2)}) - (np_1 + z_\beta\sqrt{np_1(1-p_1)}) \\
&= n(p_2 - p_1) - z_\beta(\sqrt{p_2(1-p_2)} + \sqrt{p_1(1-p_1)})\sqrt{n} \\
&> n(p_2 - p_1) - z_\beta\sqrt{n}
\end{aligned}$$

with probability determined by the quantile z_β. Note that $p_2 - p_1 > 0$, so $n(p_2 - p_1) - z_\beta\sqrt{n}$ is positive for sufficiently large n. The number of flips then required to identify the biased coin is then determined by

$$n(p_2 - p_1) - z_\beta\sqrt{n} > \frac{\log\left(\frac{1}{\alpha} - 1\right)}{|\log\left(\frac{1}{p_1} - 1\right) - \log\left(\frac{1}{p_2} - 1\right)|}.$$

Choosing a quantile z_β, substituting for p_1, p_2 and α, and then solving for n gives a lower bound for the number of flips required for distinguishing the fair coin from the biased coin.

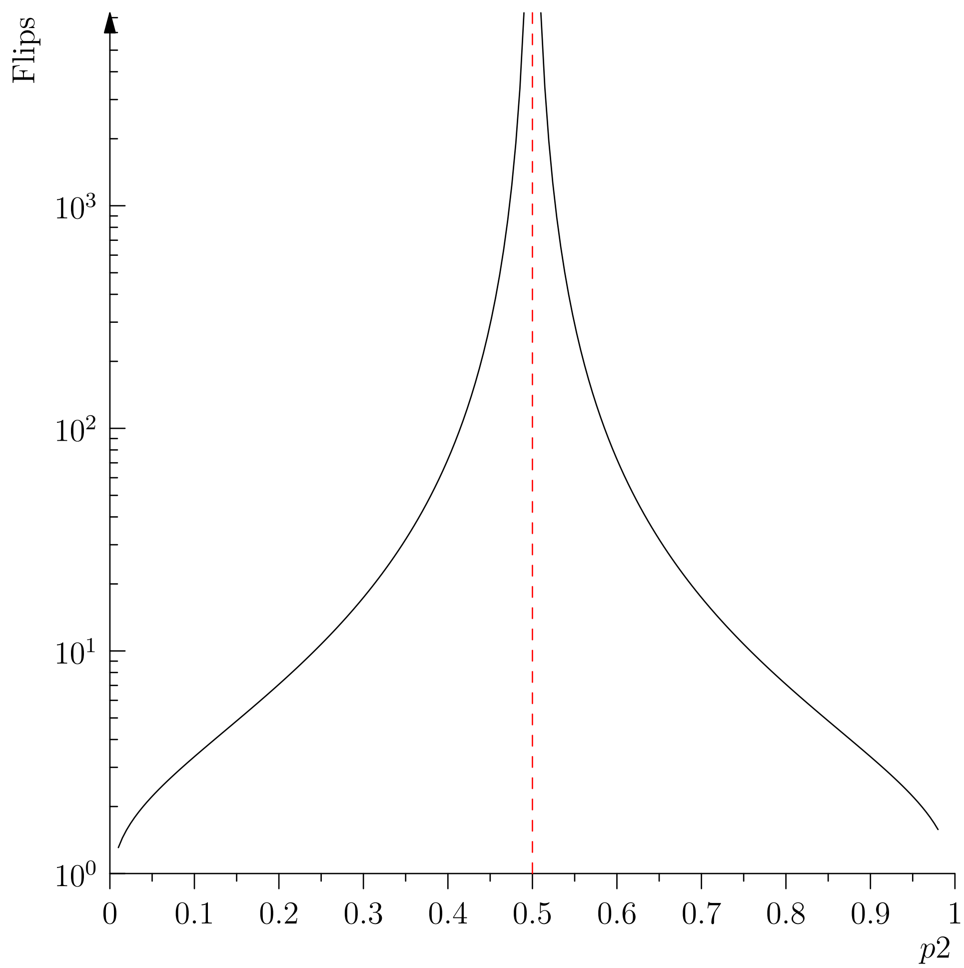


Figure 4: Number of flips required as a function of p_2 for fixed $p_1 = 0.5$ and $\alpha = 0.05$.

For example, choosing $z_\beta = 1.96$ gives the approximate probability $\beta = 0.975$ that k_L is in the range $[0, np_1 + z_\beta\sqrt{np_1(1-p_1)}]$ and independently that k_R is in the range $[np_2 - z_\beta\sqrt{np_2(1-p_2)}, n]$. Then the probability of both events is $(0.975)^2 \approx 0.95$. Using $p_1 = 0.5$, $p_2 = 0.6$ and $\alpha = 0.05$, the inequality is approximately

$$0.1n - 1.96\sqrt{n} > 7.26.$$

Solving for n gives n 519.20. As another example set $z_\beta = 0.559$ with an approximate probability of 0.712 of k_L in the range $[0, np_1 + z_\beta\sqrt{np_1(1-p_1)}]$ and independently that k_R in the range $[np_2 - z_\beta\sqrt{np_2(1-p_2)}, n]$ and joint probability of 0.507. Then solving $0.1n - 0.559\sqrt{n} > 7.26$ gives $n = 138.35$. As a last example, choosing $z_\beta = 0$ gives a probability of 0.5 of k_L in the range $[0, np_1]$ and independently with probability approximately 0.5 that k_R in the range $[np_2, n]$ and joint probability of 0.25. This is equivalent to the original calculation giving the lower bound of $n = 72.6$.

The extended Bayesian analysis introduces another parameter z_β. Using two parameters z_{β_1} and z_{β_2}, as long as $np_1 + z_{\beta_1}\sqrt{np_1(1-p_1)} < np_2 - z_{\beta_2}\sqrt{np_2(1-p_2)}$ for reasonable values of n, might be better. Now some experimentation, including binary search, is necessary to choose a value of $n \geq 73$ required to determine the biased coin with certainty 0.95. The scripts below show the results of that experimentation and compare it to the other determination methods based on the majority and the Central Limit Theorem. For example, with $p_1 = 0.5$ in the left hand, and $p_2 = 0.6$ in the right hand it takes 189 flips to indentify the fair coin in the left hand with 95% certainty. A close look at the results show why the log-likelihood Bayesian criterion requires more flips for the same certainty. The short answer is that the log-likelihood is a stricter requirement than the majority requirement in the other two approaches. In one trial with 200 flips, the fair coin in the left hand came with 108 heads while the biased coin in the right hand came up with 110 heads. The majority of heads was in the right hand, but the absolute difference of the log-likelihoods is 0.8109302, not greater than $\log(1/\alpha - 1) = 2.944439$. Both heads totals are well within the two-standard deviation range around the respective expected values of 100 heads and 120 heads respectively.

An R script testing all three methods against placing the fair and biased coins in the left and right randomly is in the scripts section. In summary, even with more flips, the Likelihood method is slightly less effective than the two majority methods. It is also a more complicated procedure.

Analysis of All Possible Algorithms

The Weak Law of Large Numbers says the probability the biased coin will have a majority of heads increases to 1 as the number of flips increases. Thus using an unlimited number of tosses will determine the biased coin. But efficiency asks for a necessary number of tosses to determine the biased coin with a satisfactory confidence. The discussion above gives several algorithms each with some number of tosses, often a large number, required to determine the biased coin with some confidence. But the question remains, are these the best algorithms? That is, among all possible algorithms is there a minimum number of tosses necessary to determine the biased coin with a certain confidence? The goal of this subsection is to get a theoretical lower bound on the necessary number of tosses.

Let $X_1, X_2, X_3, \dots$ be a sequence of i.i.d. Bernoulli random variables with $p = 1/2$ and let $Y_1, Y_2, Y_3, \dots$ be an independent sequence of i.i.d. Bernoulli random variables with $p = 1/2 + \epsilon$. Let $0 < \alpha < 1/2$ be the error tolerance or equivalently $1/2 < 1 - \alpha < 1$ is the confidence. Let $\mathcal{A}$ be an algorithm, that is, a function which takes a binary string $Z = (Z_1, Z_2, Z_3, \dots, Z_n)$ as input and outputs 0 if the algorithm determines $Z_i \sim X_i$ and outputs 1 if the algorithm determines $Z_i \sim Y_i$. Let $\mathbb{P}^m$ be the distribution of $X_1, X_2, X_3, \dots, X_m$ and $\mathbb{Q}^m$ be the distribution of $Y_1, Y_2, Y_3, \dots, Y_M$. The goal is to find a value m so that $\mathbb{P}\left[\mathcal{A}(Z) = 0\right] \ge 1 - \alpha$ and $\mathbb{Q}\left[\mathcal{A}(Z) = 1\right] \ge 1 - \alpha$. In words, find the value of m necessary to make the $\mathbb{P}^m$-measure of $\mathcal{A}(Z) = 0$ large and simultaneously make the $\mathbb{Q}^m$-measure of $\mathcal{A}(Z) = 1$ large.

Recall the **Total Variation distance** of $\mathbb{P}^m$ from $\mathbb{Q}^m$ is

$$\|\mathbb{P}^m - \mathbb{Q}^m\|_{TV} = \max_{A \subset \Omega} |\mathbb{P}^m(A) - \mathbb{Q}^m(A)|.$$

So now the goal is to find a value m so that $\|\mathbb{P}^m - \mathbb{Q}^m\|_{TV} \ge 1 - 2\alpha$. Pinsker's Inequality says that for measures μ and ν

$$D_{KL}(\mu \parallel \nu) \ge 2 \left(\|\mu - \nu\|_{TV}\right)^2$$

where $D_{KL}(\mu \parallel \nu)$ is the Kullback-Liebler Divergence of μ compared to ν.. The Kullback-Liebler Divergence satisfies the identity $D_{KL}(\mu^m \parallel \nu^m) = m D_{KL}(\mu \parallel \nu)$, sometimes called the "chain rule".

Applying these notions to the case of distinguishing the coins,

$$mD_{KL}(\mathbb{P} \parallel \mathbb{Q}) = D_{KL}(\mathbb{P}^m \parallel \mathbb{Q}^m) \ge \|\mathbb{P}^m - \mathbb{Q}^m\|_{TV} \ge 2(1-2\delta)^2.$$

Solving for m

$$m \ge \frac{2(1-2\delta)^2}{D_{KL}(1/2 \parallel 1/2+\epsilon)}$$

where $D_{KL}(1/2 \parallel 1/2+\epsilon)$ is a memorable shorthand for $D_{KL}(\mathbb{P} \parallel \mathbb{Q})$.

Now expand the definition of $D_{KL}(1/2 \parallel 1/2+\epsilon)$:

$$\begin{aligned}
&= -\frac{1}{2}\left(\log\left(1-2\epsilon\right) + \log\left(1+2\epsilon\right)\right) \\
&= -\frac{1}{2}\log(1-4\epsilon^2) \\
&= -\frac{1}{2} \cdot -4\epsilon^2 = 2\epsilon^2
\end{aligned}$$

(Check this again, since it is better than the notes.)

Now combining this estimate with the inequality for m

$$m \ge \frac{2(1-2\delta)^2}{2\epsilon^2} = (1-2\delta)^2\epsilon^{-2}.$$

The requirement that $\delta < 1/4$ guarantees that $(1-2\delta)^2 > 1/4$. However the main conclusion is that the lower bound for the number of flips is of the order of ϵ^{-2}.

The next step demonstrates an algorithm that achieves this bound, up to multiplicative constants. The proof uses the Chernoff bounds to put an upper bound on m of the same order. This means that there exists an algorithm $\mathcal{A}$ which achieves the δ error tolerance given m flips where m is of the order ϵ^{-2}.

The Chernoff bound for a binomial random variable $W \sim \mathrm{Bin}(n,p)$ and $0 < \alpha < p$, then

$$\mathbb{P}\left[W < n\alpha\right] \le \mathrm{e}^{-nD_{KL}(1/2 \| 1/2+\epsilon)}.$$

The Chernoff bound applied to a binomial random variable is the same as the Large Deviation Bound for a binomial random variable, see the definition of $h_+(\epsilon)$ and the statement of Theorem 3 in Large Deviations in the section on Large Deviations.

Define algorithm $\mathcal{A}$ taking $Z = Z_1, Z_2, \ldots, Z_m$ as input by

$$\mathcal{A}(Z) = \begin{cases} 1 & \text{(fair coin) if the numer of Heads is less than } \frac{1}{2} + \frac{\epsilon}{2} \\ 0 & \text{(biased coin) if the numer of Heads is greater than } \frac{1}{2} + \frac{\epsilon}{2} \end{cases}.$$

By the Chernoff bounds

$$\mathbb{P}\left[X_1 + X_2 + \cdots + X_m \le (\frac{1}{2} + \frac{\epsilon}{2})\right] \le \mathrm{e}^{-mD_{KL}(\frac{1}{2}+\frac{\epsilon}{2}\|\frac{1}{2})}$$
$$\mathbb{P}\left[Y_1 + Y_2 + \cdots + Y_m \le (\frac{1}{2} + \frac{\epsilon}{2})\right] \le \mathrm{e}^{-mD_{KL}(\frac{1}{2}+\frac{\epsilon}{2}\|\frac{1}{2})}.$$

The goal is to have both of these bounds less than α, so it resolves to solving both inequalities for m. The first inequality becomes

$$m \ge \frac{\log(1/\alpha)}{D_{KL}(\frac{1}{2} + \frac{\epsilon}{2} \parallel \frac{1}{2})}.$$

Likewise the second inequality also becomes

$$m \ge \frac{\log(1/\alpha)}{D_{KL}(\frac{1}{2} + \frac{\epsilon}{2} \parallel \frac{1}{2})}.$$

As shown above $D_{KL}(\frac{1}{2} + \frac{\epsilon}{2} \parallel \frac{1}{2}) \le 2\epsilon^{-2}$ so choosing m at least $2\log(1/\alpha)\epsilon^{-2}$ suffices.

Note the difference in the bounds for ϵ fixed. The lower bound grows like $(1 - 2\alpha)^2$ while the upper bound grows like $\log(1/\alpha)$, For the specific examples considered at the beginning, $\alpha = 0.05$ and $\epsilon = 0.10$, the lower bound is

$$(1 - 2\delta)^2\epsilon^{-2} = (1 - 2 \cdot 0.05)^2 \cdot (0.10)^{-2} = 81$$

and the upper bound is

$$2\log(1/\alpha)\epsilon^{-2} = 2\log(1/0.05)(0.10)^{-2} \approx 599.15.$$

The algorithms described above fall well within this range, closer to the lower bound.

Sources

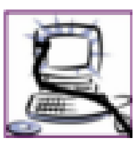
The definition of biased coin is from Wikipedia.

The simple probability problem is adapted from Bogolmony, Cut-the-Knot, Two Coins, One Fair, One Biased. The same problem appears in FlyingColoursMaths Two Coins, One Fair, One Biased.

The problem of how many tosses to find the biased coin appeared in the FiveThirtyEight.com weekly Riddler puzzle column on September 29, 2017. This first subsection with the majority solution is adapted from a white paper "What's Past is *Not* Prologue" by James White, Jeff Rosenbluth, and Victor Haghani from Elm Partners Investing, [**?**]. In turn, their example is motivated by a paper "Good and bad properties of the Kelly criterion" by McLean, Thorp and Ziemba, [**?**]. The second subsection with the statistical solution is original.

The Bayesian analysis is adapted and extended from Lessard,Finding the biased coin.

The necessary bounds for an arbitrary algorithm are adapted from lecture notes by Austin Eide.



Algorithms, Scripts, Simulations

Algorithm

Test Biased Coin Algorithms

```
    Comment Post: Determine the percentage of correct guesses of
    each of the Majority of Heads, Central Limit Theorem, and
    Log Likelihood algorithms.
Set the number of flips per experiment and number of experiments
Set the confidence level
Set the theoretical values required by each algorithm
Choose randomly which hand contains the fair and biased hand
In each hand make the number of flips for each experiment
Make a cumulative total of heads in each hand over experiments
Subtract cumulative totals of heads in Right from the Left
Calculate mean difference along the flips for each experiment
At the theoretical values for Majority and CLT, total how many experiments satisfy conditio
At the theoretical value for the Log-Likelihood, calculate the log-likelihood of outcomes in ea
For each experiment, Evaluate if the log-likelihood ratio satisfies the tolerance
For each algorithm, calculate the percent of experiments which correctly identify the biased c
Report the
```

Majority Binary Search

OPROB

bivarBinom $= \text{dbinom}([0:n], n, 0.5) \otimes \text{dbinom}([0:n], n, p)$

Bivariate binomial probability distribution on $[0:n] \times [0:n]$

Sum the probabilities on the strictly upper triangular portion

return the probability **for** each of three representative certainty values $\alpha = 0.95, 0.9, 0.75$

  Start with arbitrary high value *high* $= 1000$

  for probability p from 0.55 to 0.99 by 0.01

  Start with low value *low* $= 1$

  For efficiency, previous *high* can be used as the upper bound since PROB(n, p) is increasi

  while *high* and *low* differ by more than 1

    Set a new search value *try* at the midpoint of *high* and *low*.

    Evaluate PROB(try, p)

    if PROB$(try, p) \leq \alpha$

      low $=$ *try* using that PROB(n, p) is increasing in n

    else *high* $=$ *try* using that PROB(n, p) is increasing in n

    Record search results in vector *reqN*

For each level of certainty, draw a graph of the required number of flips versus probability p.

Statistical Binary Search

PROB

Create trinomial distribution*D1* on $[-1, 0, 1]$

Create the nth convolution power *Dnn* theoretically distributed on $[-n : n]$

The R library DISTR ignores points with probability less than 10^{-16} and so these points are n

Find the index *end0* of the last support point with probability greater than 10^{-16}

Find the index *start0* of the support point corresponding to 0

Create a vector of the probability distribution of all points with probability greater than 10^{-}

return the sum of all probabilities from [*start0* : *end0*] **for** each of three representative certa

    Start with arbitrary high value *high* = 150

    for probability p from 0.55 to 0.99 by 0.01

    Start with low value *low* = 1

    For efficiency, previous *high* can be used as the upper bound since PROB(n, p) is increasi

    while *high* and *low* differ by more than 1

        Set a new search value *try* at the midpoint of *high* and *low*.

        Evaluate PROB(try, p)

        if PROB$(try, p) \leq \alpha$

            low = *try* using that PROB(n, p) is increasing in n

        else *high* = *try* using that PROB(n, p) is increasing in n

        Record search results in vector *reqN*

For each level of certainty, draw a graph of the required number of flips versus probability p.

Scripts

R R script for Testing Algorithms.

```r
n <- 200   # number of flips
k <- 500   # number of experiments
alpha <- 0.05

NMaj <- 143  # required for Majority
NCLT <- 143  # required for CLT
NLL <- 189   # required for logLikelihood

# choices <- rep(FALSE, k)
choices <- (runif(k) <= 0.5)
pLeft <- ifelse(choices, 0.6, 0.5)
pRight <-  ifelse(!choices, 0.6, 0.5)

```

```
ULeft <- array( runif(k*n), dim=c(k,n) ) # flips along rows
coinFlipsLeft <- ULeft <= pLeft  # take advantage of recycling
URight <- array( runif(k*n), dim=c(k,n) ) # flips along rows
coinFlipsRight <- URight <= pRight  # take advantage of recycling

headsTotalLeft <- apply(coinFlipsLeft, 1, cumsum)
headsTotalRight <- apply(coinFlipsRight, 1, cumsum)

LmR <- headsTotalLeft - headsTotalRight
CLT <- LmR/(1:n)

guessBiasedMajority <- LmR[NMaj,] > 0
guessBiasedCLT <- CLT[NCLT, ] - 0.1 > -0.1

logQL <- dbinom(headsTotalLeft[NLL, ], NLL, 0.6, log = TRUE) +
         dbinom(headsTotalRight[NLL, ], NLL, 0.5, log = TRUE)
logQR <- dbinom(headsTotalRight[NLL, ], NLL, 0.6, log = TRUE) +
         dbinom(headsTotalLeft[NLL, ], NLL, 0.5, log = TRUE)
rhs <- log(1/alpha - 1)
guessBiasedLL <- ( (logQL - logQR) > rhs )

correctMajority <- sum(guessBiasedMajority == (pLeft == 0.6))
perCentCorrectMajority <- 100 * correctMajority/k
perCentCorrectMajority

correctCLT <- sum(guessBiasedCLT == (pLeft == 0.6))
perCentCorrectCLT <- 100 * correctCLT/k
perCentCorrectCLT

correctLogLike <- sum(guessBiasedLL == (pLeft == 0.6))
perCentCorrectLL <- 100 * correctLogLike/k
perCentCorrectLL

cat("Majority of Heads Algorithm correct:", perCentCorrectMajority, "\n")
cat("Central Limit Theorem Algorithm correct:",
```

```
    perCentCorrectCLT, "\n")
cat("Log-Likelihood Algorithm correct:", perCentCorrectLL, "\n")
```

R script for Majority Binary Search.

```
prob <- function(n, p) {
    fair <- dbinom(0:n, n, 0.5)          #fair binomial distribution
    biased <- dbinom(0:n, n, p)          #biased binomial dist
    bivarBinom <- fair %o% biased        #outer product, (n+1) x (n+1) matrix
    bivarBinom[ lower.tri(bivarBinom, diag=TRUE) ] <- 0
    # retain only the upper triangular values, without diagonal
    sum( bivarBinom )                    #add them all up
}

reqN <- c()                              #initialize vector to hold results

for (certainty in c(0.95, 0.9, 0.75)) { #three representative values
    high <- 1000                         #arbitrary guess for max for search
    for (p in seq(0.55, 0.99, length=45)) { #array of bias values
        low <- 1                             #minimum to start search

        while (high - low > 1) {
            try <- floor( (low + high)/2 ) #binary search
            if ( prob(try, p) <= certainty ) low <- try else high <- try
            ## uses monotone increasing nature of probability
        }

        reqN <- c(reqN, low)             #add search result to vector of results
    }
}

```

```r
N <- matrix(reqN, 45,3)                   #reformat to three columns, one for each certainty

ps = seq(0.55, 0.99, length=45)           #array of bias values to plot on horiz
plot( ps, N[,1], pch=20, xlab="p", ylab="N") #plot required N for certainty 0.95
points( ps, N[,2], pch=20, col="red")         #add plot for required N for 0.90
points( ps, N[,3], pch=20, col="blue")        #add plot for required N for 0.75
```

R script for Statistical Binary Search.

```r
library(distr)

prob <- function(nn, p) {
    D1 <- DiscreteDistribution( -1:1, c((1-p)/2, 1/2, p/2) )
    ## trinomial distribution of X_n = L_n - R_n
    Dnn <- convpow(D1, nn)
    ## distribution of sum_{j=1}^n X_n
    end0 <- NROW( (support(Dnn)) )
    ## last non-zero support point
    start0 <- match( 0, (support(Dnn))) + 1
    ##  index of support point of 0, then add 1
    distDnn <- d(Dnn)(support(Dnn))
    ## extract vector of distribution values on [start0, end0]
    sum( distDnn[start0:end0] )          #add up the probability
}

reqN <- c()                               #initialize vector to hold results

for (certainty in c(0.95, 0.9, 0.75)) { #three representative value
    high <- 150                            #place to start search
    for (p in seq(0.55, 0.99, length=45)) { #array of bias values
        low <- 1                             #minimum to start search
```

```
        # high <- high from previous         #a maximum to start with
        ## For greater efficiency, note that the required number of flips
        ## for the next p  will be not more than the number of flips for previous
        ## p, so restrict search to interval n=1 to to n=reqN from previous search
        while (high - low > 1) {
            try <- round( (low + high)/2 ) #binary search
            if ( prob(try, p) <= certainty ) low <- try else high <- try
        }

        reqN <- c(reqN, high)              #add search result to vector of results
    }
}

N <- matrix(reqN, 45,3)                    #reformat to three columns, one for each certainty

ps = seq(0.55, 0.99, length=45)            #array of bias values to plot on horiz
plot( ps, N[,1], pch=20, xlab="p", ylab="N") #plot for required N for certainty 0.95
points( ps, N[,2], pch=20, col="red")        #add plot for required N for 0.90
points( ps, N[,3], pch=20, col="blue")       #add plot for required N for 0.75
```

R script for Algorithms for Biased Coins.

```
alpha <- 0.05
alphaRatio <- (1-alpha)/alpha
logAlphaRatio <- log((1-alpha)/alpha)

p1 <-  0.5
p2 <- 0.6

n <- 200
k <- 100

# uniform fair in Left hand
```

```
coinFlipsLeft <- array( (runif(n*k) <= p1), dim=c(k,n))
coinFlipsRight <- array( (runif(n*k) <= p2), dim=c(k,n))

headsTotalLeft <-  apply(coinFlipsLeft, 1, sum)
headsTotalRight <-  apply(coinFlipsRight, 1, sum)
# see FEATURES for why cumsum, not just column sum

logQkleft <-  dbinom(headsTotalLeft, n, p1, log=TRUE) +
    dbinom(headsTotalRight, n, p2, log=TRUE)
logQkright <-  dbinom(headsTotalRight, n, p1, log=TRUE) +
    dbinom(headsTotalLeft, n, p2, log=TRUE)

logLikeRatio <- abs(logQkleft - logQkright)
logGuessBiased <- logLikeRatio > logAlphaRatio

logcorrect <- sum(logGuessBiased == (p1 == 0.5))
logperCentCorrect <- 100 * logcorrect/k
logperCentCorrect
```

Problems to Work for Understanding

1: You have two coins. One is fair with $\mathbb{P}[H] = \frac{1}{2}$. The other coin is biased with $\mathbb{P}[H] = p > \frac{1}{2}$. First you toss one of the coins once, resulting in heads. Then you toss the other coin three times, resulting in two heads. What is the least value of p that will allow you to decide which is the biased coin based on this information?

2: Show that the log-likelihood expression with the binomial probabilities simplifies to

$$|k_L - k_R| \cdot |\log\left(\frac{1}{p_1} - 1\right) - \log\left(\frac{1}{p_2} - 1\right)| > \log\left(\frac{1}{\alpha} - 1\right).$$

Reading Suggestion:

Outside Readings and Links:

1.
2.
3.
4.

Solutions

1: Interestingly, the probability of two heads in three tosses from a fair coin and one head in one toss from a biased coin with $p > 1/2$, specifically $\frac{3p}{8}$, is always greater, for any p, than the probability of two heads in three tosses from a biased coin and one head in one toss from a fair coin, specifically $\frac{3}{2}p^2(1-p)$.

2: Start with

$$\frac{Q(k \mid \theta_L)}{Q(k \mid \theta_R)} = \left(\frac{p_1}{p_2}\right)^{k_L - k_R} \cdot \left(\frac{1-p_2}{1-p_1}\right)^{k_L - k_R}.$$

Note that the binomial coefficients are the same in numerator and denominator so they cancel. Take logarithms of both sides, use laws of exponents

$$\log(Q(k \mid \theta_L)) - \log(Q(k \mid \theta_R)) = (k_L - k_R) \log\left(\left(\frac{p_1}{p_2}\right) + (k_L - k_R) \log\left(\frac{1-p_2}{1-p_1}\right)\right).$$

Factor and rearrange

$$\log(Q(k \mid \theta_L)) - \log(Q(k \mid \theta_R)) = (k_L - k_R)\left(\log(p_1) - \log(p_2) + \log(1-p_2) - \log(1-p_1)\right).$$

Rearrange again

$$\begin{aligned} \log(Q(k \mid \theta_L)) - \log(Q(k \mid \theta_R)) &= (k_L - k_R)\left(\log\left(\frac{p_1}{1-p1}\right) - \log\left(\frac{p_2}{1-p_2}\right)\right) \\ &= (k_L - k_R)\left(\log\left(-\frac{1-p1}{p_1}\right) + \log\left(\frac{1-p_2}{p_2}\right)\right) \end{aligned}$$

Expanding the fractions and taking absolute values gives the desired right hand side

$$|k_L - k_R| \cdot |\log\left(\frac{1}{p_1} - 1\right) - \log\left(\frac{1}{p_2} - 1\right)| > \log\left(\frac{1}{\alpha} - 1\right).$$

Steve Dunbar's Home Page, `http://www.math.unl.edu/~sdunbar1`

Email to Steve Dunbar, `sdunbar1 at unl dot edu`

Last modified: Processed from LaTeX source on October 13, 2020